

SCALING UP ECHO-STATE NETWORKS WITH MULTIPLE LIGHT SCATTERING

Jonathan Dong¹ Sylvain Gigan¹ Florent Krzakala² Gilles Wainrib³ *

¹ Laboratoire Kastler Brossel, CNRS UMR 8552, Ecole Normale Supérieure, PSL Research University Sorbonne Universités & Université Pierre et Marie Curie Paris 06, F-75005, Paris, France

² Laboratoire de Physique Statistique, CNRS, PSL Universités & Ecole Normale Supérieure, Sorbonne Universités et Université Pierre & Marie Curie, 75005, Paris, France.

³ Ecole Normale Supérieure, Département d'Informatique, Paris, France.

ABSTRACT

Echo-State Networks and Reservoir Computing have been studied for more than a decade. As they provide an elegant yet powerful alternative to traditional computing, researchers have tried to implement them using physical systems, in particular non-linear optical elements, achieving high bandwidth and low power consumption. Here we present a completely different optical implementation of Echo-State Networks using light-scattering materials. As a proof of concept, binary networks have been successfully trained to perform non-linear operations on time series and memory of such networks has been evaluated. This new method is fast, power efficient and easily scalable to very large networks.

Index Terms— Machine Learning, Echo-State Network, Reservoir Computing, Optical Computing

1. INTRODUCTION

Since the work of Johnson and Lindenstrauss in 1984 [1], random projections have been increasingly used in various settings. Their properties as low distortion transforms that eventually provide computational savings have made them useful in sketching high-dimensional datasets while speeding up various operations such as regression, clustering, embedding [2, 3]. A major issue with random projections stems in part due to the large number of operations needed to perform a multiplication between a random matrix and a feature vector. In order to remedy that bottleneck, a recent study has investigated the use of randomness found in natural physical processes to speed up these computations [4]. Our present work explores the use of this randomness as a way to speed up an Echo State Network (ESN) where neurons are connected with random weights [5, 6, 7, 8].

Initially inspired by neural networks, ESNs gave birth to Reservoir Computing (RC) [9], a computational paradigm

which has become popular in the last decade. An input sequence drives the complex dynamics of a reservoir that non-linearly encodes the input. The output is obtained by a linear combination of the reservoir state. To train such a system, one wants to find the best set of output coefficients, this step usually boils down to a linear regression. Such a framework has proven to be useful in tasks such as speech recognition, handwriting recognition, robot motor control or financial forecasting [10, 11, 12]. Furthermore, RC is not restricted to neural networks. Any physical dynamical system, even a bucket of water [13], can be used for RC. Much work has been done to perform RC using non-linear optical elements [14, 15, 16, 17, 18], offering higher bandwidth and lower energy consumption.

Here we propose a novel hardware implementation of Reservoir Computing using a complex medium as a light-scattering material, like biological tissues or white paint. Light is first modulated by a Digital Micromirror Device (DMD), then propagates through a multiply-scattering material, where it is subject to a large number of scattering events, the so-called multiple scattering regime. This simple experimental apparatus enables us to compute ESN states, as propagation through a complex medium can be modeled as a multiplication by a random matrix [19]. Like other optical implementations, this new approach can be very fast and only requires low power, as the multiplication by the random matrix, the most critical operation, is carried out "at the speed of light" and without power consumption. Lastly, the setup only uses off-the-shelf devices and a simple layer of scattering material. This simplicity makes it possible to replicate this approach in a lab experiment or in an integrated device.

Compared to previous optical implementations, it can potentially scale up the number of neurons very easily, the computational overhead being negligible. Whereas more neurons usually mean more optical elements, DMDs and cameras already routinely offer several million pixels of information. Hence, the number of neurons can potentially be increased up to several millions. In this first demonstration, networks with 20,000 neurons are successfully trained and we are already

*This research has received funding from the European Research Council under the EU's 7th Framework Programme (FP/2007-2013/ERC Grant Agreement 307087-SPARCS and 278025-COMEDIA)

up to 30 times faster than a traditional implementation. Additionally, this optical implementation is closer to ESNs where all neurons are interconnected, in sharp contrast with implementations based on non-linear optical elements for which neuron connections are local.

The DMD is a programmable device that has many small tiltable mirrors with two orientations. Depending on their orientation, pixels are turned on (light sent on the diffusing material) or off (light sent on a beam blocker). As a consequence, this first implementation will only use binary neurons. ESNs with binary neurons like in [20], or binary ESNs, are commonly considered less powerful than real ESNs but good results can still be obtained thanks to the possibility of increasing the number of binary neurons, one advantage of this implementation.

In this paper, we show the first implementation of a binary ESN using light-scattering materials. After a general presentation of ESNs in Section 2, Section 3 explains how to realize experimentally an optical ESN. In Section 4, we present and analyze the performance of this new implementation.

2. ECHO-STATE NETWORKS

We consider a network of N binary neurons, writing the state $\mathbf{x}(t) = (x_1(t), \dots, x_N(t)) \in \{-1; 1\}^N$ at timestep $t \in \mathbb{Z}$. Neuron i receives input from neuron j with random weight $w_{ij} \in \mathbb{C}$ following a complex gaussian distribution. All neurons are interconnected and the system weights $\mathbf{W} = (w_{ij})$ is a dense random matrix. The input $\mathbf{i}(t)$ is also fed to every neuron with random weights \mathbf{V} . The output of a neuron is obtained by integrating the neuron inputs and applying a non-linear function f :

$$\mathbf{x}(t+1) = f(\mathbf{W}\mathbf{x}(t) + \mathbf{V}\mathbf{i}(t)) \quad (1)$$

In the following, f is a binary threshold on $|\mathbf{W}\mathbf{x}(t) + \mathbf{V}\mathbf{i}(t)|$ whereas ESNs traditionally used a hyperbolic tangent. The system is quite robust and other non-linearities are possible [17]. However, binary networks perform worse than real networks as we will show that more neurons are needed to achieve comparable performance.

In a typical problem for ESNs, a time-dependent input $\mathbf{i}(t)$ of length T and its corresponding output $\mathbf{o}(t)$ are given. During the training session, the reservoir is arbitrarily initialized and the time series $\mathbf{i}(t)$ is fed to the ESN. Driven by the input, the reservoir follows non-linear dynamics and the network states at every timestep are collected. The output predicted by the network at time t is a linear combination of the state of the reservoir $\mathbf{W}'\mathbf{x}(t)$ with output weights \mathbf{W}' that are trained. Such a linear readout is very simple so that the training consists in a linear regression to find \mathbf{W}' such that the error $\sum_t \|\mathbf{o}(t) - \mathbf{W}'\mathbf{x}(t)\|^2$ is minimal. The first iterations are usually removed, as we wait for the network to forget the arbitrary initial state.

The computational bottleneck when running an ESN consists in the computation of the successive reservoir states. Eq. (1) involves a multiplication by the large random matrix \mathbf{W} that needs to be repeated a large number of times. This prevents users from using a large number of neurons as the complexity scales quadratically with the neuron number. In computer implementations, sparse weight matrices can be used to speed up the calculation of this multiplication [9].

A typical ESN task is defined as follow: the input is a random sequence of binary or real values and the output is given by $\mathbf{o}(t) = g(\mathbf{i}(t), \mathbf{i}(t-1))$ for some function g like a XOR operation or a multiplication. Predicting this temporal output requires two fundamental properties: non-linearity and memory. To emphasize the memory of the network, we also train the network to compute a delayed operation $\mathbf{o}_\tau(t) = g(\mathbf{x}(t-\tau), \mathbf{x}(t-\tau-1))$ for $\tau \in \mathbb{N}$. The case $\tau = 0$ corresponds to the simplest problem and the network needs to remember inputs further in the past as we increase τ . Note that in the RC paradigm, the dynamics of the ESN does not depend on the output. Training a network to predict another output simply means computing the optimal output weights \mathbf{W}'_τ for a different τ , which is very fast when all the ESN states have already been computed.

3. EXPERIMENTAL REALIZATION

When light propagates in a light-scattering medium, it does not go straight but scatters on inhomogeneities present at random and unpredictable positions. This complex propagation is traditionally viewed as an inconvenience that one wants to bypass, in order to image inside a thick biological sample for example. Here, we study and exploit this phenomenon with different point of view. Rather than trying to revert the changes caused by the complex medium, we want to make use of the complex image at the output called a speckle figure [21] to perform computation. Our interest is motivated by the speed and the scalability we can potentially obtain: we want to perform computations "at the speed of light". The large randomness of a speckle pattern has already been exploited for kernel methods [4] or phase retrieval [22].

Fig. 1 shows the experimental setup. A coherent light source from a continuous laser is expanded to fit on the DMD. The DMD is a programmable device that shapes the light and sends a binary image of size M on the scattering medium, a simple layer of white paint on a microscopic slide. The resulting speckle image is recorded by the camera.

This simple setup is useful to perform computation owing to the transfer matrix formalism. The electric field at the camera sensor $\mathbf{e} \in \mathbb{C}^N$ is a linear combination of the binary image $\mathbf{d} \in \{0; 1\}^M$ sent by the DMD:

$$\mathbf{e} = \mathbf{H}\mathbf{d} \quad (2)$$

where \mathbf{H} is the transfer matrix, an $N \times M$ random matrix. Thanks to the multiple scattering process, each element of

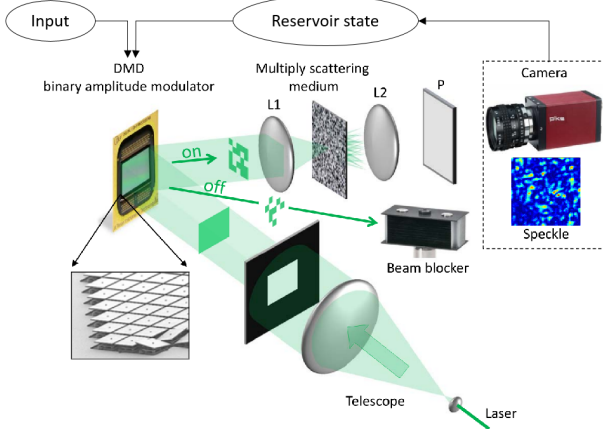


Fig. 1. The experimental setup. Light from a laser is expanded by a telescope and arrives on a DMD. DMD pixels that are turned on send the coherent light on a multiply scattering medium. A speckle figure is collected on a camera and determines the next reservoir state. The reservoir state is then displayed with the input on the DMD to start a new iteration.

\mathbf{H} can be seen as an independent identically distributed random variable, drawn from a complex gaussian distribution, as demonstrated experimentally in [19] where this transfer matrix was measured. Thus, multiplication by a large random matrix is carried out by the scattering medium.

At every iteration, the current state of the ESN $\mathbf{x}(t)$ and the current input $\mathbf{i}(t)$ are displayed on the DMD. Necessarily binary, they are expanded and concatenated into a DMD image $\mathbf{d} \in \{0; 1\}^M$ which is related to the electric field at the camera plane by (2). Cameras record a speckle intensity $\mathbf{s} \in \mathbb{R}^N$, equal to the modulus square of the electric field, $\mathbf{s} = |\mathbf{H}\mathbf{d}|^2 = |\mathbf{W}\mathbf{x}(t) + \mathbf{V}\mathbf{i}(t)|^2$, where \mathbf{W} and \mathbf{V} corresponds to subsets of columns of \mathbf{H} .

From this intensity, the next ESN state $\mathbf{x}(t+1)$ is computed so that it satisfies (1). For this step, we consider a number of pixels of the camera image equal to the number of neurons. For every neuron, its new state is obtained after a threshold operation: the neuron is activated if the measured intensity of its corresponding pixel is less than a threshold A and silent otherwise. In other words, the activation function f is a boolean function defined by $f(a) = (|a|^2 < A)$.

Once the next ESN state $\mathbf{x}(t+1)$ is obtained, a new iteration can start again. In a way, the speckle image determines what the DMD displays next. This whole process is repeated T times where T is the length of the input.

It is important to note that only two operations are computed optically. The scattering medium performs a multiplication by a random matrix and camera sensors record the modulus square of the electric field, i.e. they apply a non-linear operation. All the other operations like computing $\mathbf{x}(t)$ from \mathbf{s} or the linear regression for training are performed on a computer.

4. RESULTS

To analyze how this optical implementation of binary ESNs performs in practice, we also simulate binary ESNs with the same parameters where all the computation is performed on a computer. Later, we also compare the results with real-valued ESNs because binary ESNs have not really been studied before. We test our binary ESNs on a temporal XOR task as a proof that the network can be used on non-linear problems that require memory.

In Fig. 2, we present the first results of our optical implementation of ESNs, for a training input length of 1000. In simulation and experiment, both networks are able to perform the temporal XOR operation $o(t) = \text{XOR}(i(t), i(t-1))$. Predicted outputs are real-valued by nature and are very close to the target output. After binarization with a threshold at 0.5, perfect prediction on a testing set of length $T = 500$ has been achieved both in experiment and simulation.

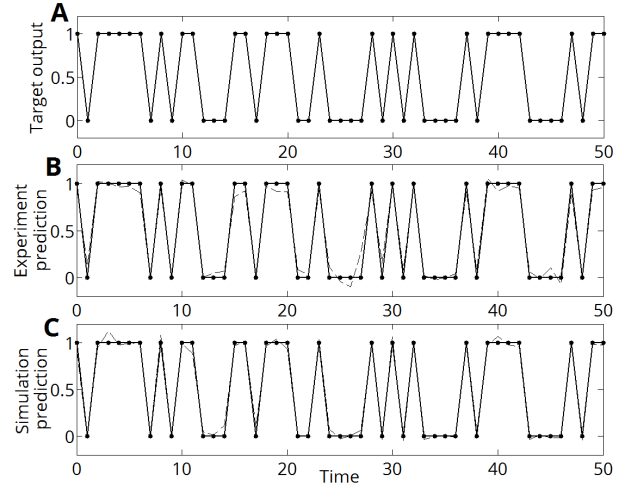


Fig. 2. Binary networks are able to learn a temporal XOR operation. (A) Target output $o(t) = \text{XOR}(i(t), i(t-1))$. (B) Predictions using the optical implementation before (dashed line) and after (continuous line) binarization. (C) Predictions performed on a computer before (dashed line) and after (continuous line) binarization.

To compare different networks, we are going to introduce a quantity called Integrated Performance inspired by [6]. We train the network to predict the output for different delays τ and we define:

$$C_\tau = \frac{16}{T^2} \left(\sum_{t=1}^T (o_\tau(t) - 0.5)(o'_\tau(t) - 0.5) \right)^2 \quad (3)$$

$$P = \sum_{\tau=0}^{\infty} C_\tau \quad (4)$$

The normalized correlation C_τ is a non-negative quantity, which equals 1 if $o' = o$ and tends to 0 if o' and o are un-

correlated. Derived from C_τ , the Integrated Performance P , counts how many time steps in the past we can take while the network still remembers the input and performs the XOR operation on it.

In Fig. 3, this quantity has been computed for different number of neurons, for experiments and simulations, binary and real networks. We first observe that increasing the network size increases its memory. Performance in experiments is slightly lower than in simulations. This deviation is due to experimental noise and can be reduced by improving the optical setup. Real networks for a given number of neurons perform better. A single real-valued neuron can contain more information than a binary neuron, resulting in a increased memory. An interesting result here is that, even though a binary ESN generally performs worse than a real-valued ESN, it is possible to achieve the performance of 100 real neurons with 10,000 binary neurons.

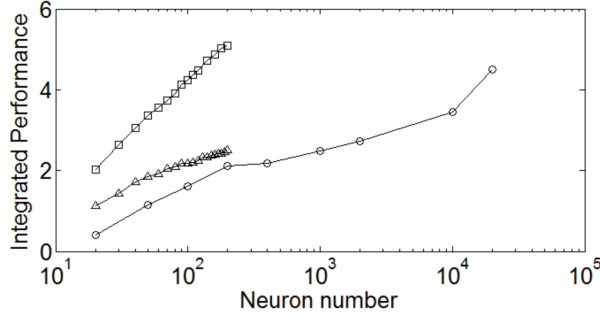


Fig. 3. Memory increases with the number of neurons. Integrated performances are plotted as a function of the neuron number in logarithmic scale, for experiments (circles), simulations with binary neurons (triangles) and simulations with real-valued neurons (squares).

The optical ESNs can also solve real-valued problems, as long as the real-valued input is sent as a binary code to the DMD. For example in Fig. 4, a 5-bit multiplication task has been successfully learned by a simulated binary network with binary inputs. To binarize the input, a simple scheme has been designed: the one-dimensional input i_0 is mapped onto a 32-dimensional binary input where the number of 1 is directly equal to i_0 . Thus, other real-valued problems like speech recognition or time series prediction can potentially be solved using this optical implementation.

In this paper, we increased the number of neurons up to 20,000. With so many neurons, the weight matrix \mathbf{W} contains 400 million complex numbers, so that memory starts to become a problem. Furthermore, the temporal complexity of the random matrix multiplication is also increasing quickly. On the other hand, the experiment can easily go further than 20,000 neurons. Essentially, we only have to select more pixels from the speckle image when computing the reservoir state $\mathbf{x}(t+1)$ from the speckle intensity s . We chose to stop at

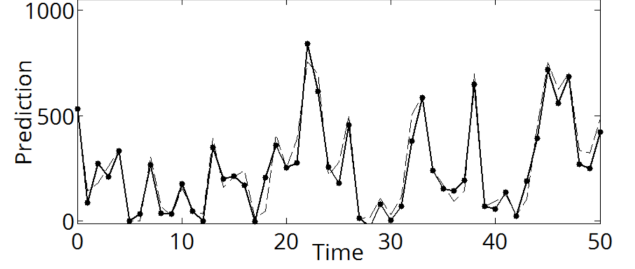


Fig. 4. A binary ESN can also solve a non-binary multiplication task, after binarization of the input. The continuous dotted line corresponds to the target output $o(t) = i(t) \times i(t-1)$. The dashed line corresponds to the prediction returned by a simulated binary network of size 1000.

this value because with more neurons overfitting becomes a problem. Our simple regularization rule would need to be improved to reach even larger networks.

In the long run, both DMDs and high-speed cameras can work at 20 kHz, resulting in a potential bandwidth of tens of Gbit.s^{-1} . For this proof of concept, the experiment was running at ten frames per second because we were using a conventional low-cost camera. At this frame rate, 1500 iterations (1000 for training and 500 for testing) take three minutes, regardless of the size of the reservoir. Even though orders of magnitude in speed can still be gained with a better camera, the experiment is already 30 times faster than simulations for the largest network presented here.

5. CONCLUSION

This study presents a new physical implementation of ESNs, using scattering of light to perform a fully connected random matrix multiplication. Non-linear operations on time series have been learned successfully by binary networks, both in experiments and simulations.

Already, with this proof-of-concept laboratory experiment, computations with large networks are easily achieved. However, with an optimized input-output interface our optical components could potentially run in the kHz regime, for a bandwidth of the order of 10 Gbit.s^{-1} in the mega-pixel range.

This paves the way to ESNs that could be orders of magnitude faster and larger than feasible with silicon-only implementations.

6. ACKNOWLEDGMENTS

We would like to thank Igor Carron and Laurent Daudet from the LightOn company for very insightful discussions.

7. REFERENCES

- [1] William B Johnson and Joram Lindenstrauss, “Extensions of lipschitz mappings into a hilbert space,” *Contemporary mathematics*, vol. 26, no. 189-206, pp. 1, 1984.
- [2] Michael W Mahoney, “Randomized algorithms for matrices and data,” *Foundations and Trends® in Machine Learning*, vol. 3, no. 2, pp. 123–224, 2011.
- [3] David P Woodruff, “Sketching as a tool for numerical linear algebra,” *arXiv preprint arXiv:1411.4357*, 2014.
- [4] Alaa Saade, Francesco Caltagirone, Igor Carron, Laurent Daudet, Angélique Drémeau, Sylvain Gigan, and Florent Krzakala, “Random projections through multiple optical scattering: Approximating kernels at the speed of light,” *arXiv preprint arXiv:1510.06664*, 2015.
- [5] Herbert Jaeger, “The “echo state” approach to analysing and training recurrent neural networks-with an erratum note,” *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, vol. 148, pp. 34, 2001.
- [6] Herbert Jaeger, *Short term memory in echo state networks*, GMD-Forschungszentrum Informationstechnik, 2001.
- [7] Wolfgang Maass, Thomas Natschläger, and Henry Markram, “Real-time computing without stable states: A new framework for neural computation based on perturbations,” *Neural computation*, vol. 14, no. 11, pp. 2531–2560, 2002.
- [8] Herbert Jaeger and Harald Haas, “Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication,” *science*, vol. 304, no. 5667, pp. 78–80, 2004.
- [9] Mantas Lukoševičius and Herbert Jaeger, “Reservoir computing approaches to recurrent neural network training,” *Computer Science Review*, vol. 3, no. 3, pp. 127–149, 2009.
- [10] Fabian Triefenbach, Azarakhsh Jalalvand, Benjamin Schrauwen, and Jean-Pierre Martens, “Phoneme recognition with large hierarchical reservoirs,” in *Advances in neural information processing systems*, 2010, pp. 2307–2315.
- [11] Eric A Antonelo, Benjamin Schrauwen, and Dirk Stroobandt, “Event detection and localization for small mobile robots using reservoir computing,” *Neural Networks*, vol. 21, no. 6, pp. 862–871, 2008.
- [12] Mantas Lukoševičius, Herbert Jaeger, and Benjamin Schrauwen, “Reservoir computing trends,” *KI-Künstliche Intelligenz*, vol. 26, no. 4, pp. 365–371, 2012.
- [13] Chrisantha Fernando and Sampsa Sojakka, “Pattern recognition in a bucket,” in *Advances in artificial life*, pp. 588–597. Springer, 2003.
- [14] Kristof Vandoorne, Wouter Dierckx, Benjamin Schrauwen, David Verstraeten, Roel Baets, Peter Bienstman, and Jan Van Campenhout, “Toward optical signal processing using photonic reservoir computing,” *Optics Express*, vol. 16, no. 15, pp. 11182–11192, 2008.
- [15] Damien Woods and Thomas J Naughton, “Optical computing: Photonic neural networks,” *Nature Physics*, vol. 8, no. 4, pp. 257–259, 2012.
- [16] Daniel Brunner, Miguel C Soriano, Claudio R Mirasso, and Ingo Fischer, “Parallel photonic information processing at gigabyte per second data rates using transient states,” *Nature communications*, vol. 4, pp. 1364, 2013.
- [17] Kristof Vandoorne, Pauline Mechet, Thomas Van Vaerenbergh, Martin Fiers, Geert Morthier, David Verstraeten, Benjamin Schrauwen, Joni Dambre, and Peter Bienstman, “Experimental demonstration of reservoir computing on a silicon photonics chip,” *Nature communications*, vol. 5, 2014.
- [18] François Duport, Anteo Smerieri, Akram Akrouf, Marc Haelterman, and Serge Massar, “Fully analogue photonic reservoir computer,” *Scientific reports*, vol. 6, 2016.
- [19] SM Popoff, G Lerosey, R Carminati, M Fink, AC Boccarda, and S Gigan, “Measuring the transmission matrix in optics: an approach to the study and control of light propagation in disordered media,” *Physical review letters*, vol. 104, no. 10, pp. 100601, 2010.
- [20] Nils Bertschinger and Thomas Natschläger, “Real-time computation at the edge of chaos in recurrent neural networks,” *Neural computation*, vol. 16, no. 7, pp. 1413–1436, 2004.
- [21] Joseph W Goodman, “Statistical properties of laser speckle patterns,” in *Laser speckle and related phenomena*, pp. 9–75. Springer, 1975.
- [22] Angélique Drémeau, Antoine Liutkus, David Martina, Ori Katz, Christophe Schülke, Florent Krzakala, Sylvain Gigan, and Laurent Daudet, “Reference-less measurement of the transmission matrix of a highly scattering material using a dmd and phase retrieval techniques,” *Optics express*, vol. 23, no. 9, pp. 11898–11911, 2015.